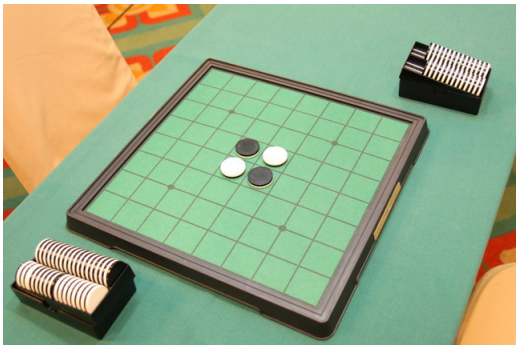
Reversi For CS107

Reversi (黑白棋) is a strategy board game for two players, played on an 8×8 uncheckered board. It was invented in 1883.



Basics

There are sixty-four identical game pieces called disks (棋子), which are light (白) on one side and dark (黑) on the other. Players take turns placing disks on the board with their assigned color facing up. During a play, any disks of the opponent's color that are in a straight line and bounded by the disk just placed and another disk of the current player's color are turned over to the current player's color. The objective of the game is to have the majority of disks turned to display one's color when the last playable empty square is filled.

Play

1. The game begins with four disks placed in a square in the middle of the grid, two facing white-side-up, two dark-side-up, so that the same-colored disks are on a diagonal. Convention has this such that the dark-side-up disks are to the north-east and south-west. The dark player moves first.
2. Play always alternates. After placing a dark disk, dark turns over (flips to dark, captures) the single disk (or chain of light disks) on the line between the new piece and an anchoring dark piece. No player can look back to the previous status of disks when playing moves. A valid move is one where at least one piece is reversed (flipped over).
3. Now light plays. This player operates under the same rules, with the roles

reversed: light lays down a light piece, causing a dark piece to flip.

4. Players take alternate turns. If one player can not make a valid move, play passes back to the other player. When neither player can move, the game ends. This occurs when the grid has filled up or when neither player can legally place a piece in any of the remaining squares. This means the game may end before the grid is completely filled. This possibility may occur because one player has no pieces remaining on the board in that player's color.
5. The player with the most pieces on the board at the end of the game wins.

Requirements

You are required to form groups of two (with the same lab teacher, no exception). Please design and implement a Java program to simulate the Reversi game for two players. Note that the only programming language you can use in this project is Java (Scala and Kotlin are accepted.)

There are four tasks below to accomplish, each of which has several points towards the final mark of a group. A framework/skeleton of the game will be release shortly to facilitate your programming.

Task 1: Initialize Game (20 pts)

- Your program should be able to initialize a new chess game, which includes the chessboard and the initial disks in their correct position.
- Your program should be able to display the status of the game (In Progress, Dark Turn or Light Turn, etc.)
- Your program should be able to restart a game by clicking a button rather than closing it and open the game again.

Task 2: Load and Save a Game (20 pts)

- Your program should be able to load an existing game from a text file with a pre-defined format by clicking buttons. After loading, all disks should be placed at their positions given in the text file. The save file includes at least the current chessboard, the previous moves, and the current side to play (dark or light).
- Your program should be able to perform error check, e.g., there is no winner yet, any move is invalid, etc.
- Your program should be able to save the current game into a text file.

Task 3: Play the game (20 pts)

- Your program should detect the winning status of the game, and end the game when there is a winner.
- Your program should allow disks to be flipped according to the rules.
- During one game, your program should be able to switch between the normal and the cheat mode, the latter of which allows a player to place either dark or light disks at any empty position.

Task 4: Graphical User Interface (20 pts)

- Your program should have a graphical user interface using Java Swing (JavaFX is accepted.)

Bonus (20/30 pts):

If your program satisfies all the above basic requirements, you will get 80 points. The remaining 20/30 points will be given as bonus. You are highly encouraged to go beyond our requirements. Below are some possible ways to get bonus. Compare to the bonus points, the basic points are easier to get. Here you need to rely on your own ability to present your programming charm!

- Design Human vs. Machine mode of different difficulty level, and make the machine player smarter.
- Design a platform for your game, such as adding multi-user, ranking list, adding start menu for selecting the game modes, etc.
- Make your game looks nicer, such as change the theme, adding sound effect, adding background music, adding more prompt label when the game is in process.
- Show possible moves when a disk is being placed.
- Play the process of placing and flipping disks after loading the save file.
- Undo operation.
- Support on-line mode in Local Area Network.
- Pack the game as an executable that can be executed on a computer without JRE.
- And more.

During project evaluation, your bonus functions will be evaluated according to their difficulty and novelty, and grouped into A/B/C/D class. A grants 8 pts, B grants 4 pts,

C grants 2 pts, D grants 1 pts. Points granted from the same class by multiple bonus functions can stack. All bonus functions count towards the bonus points you can get. This part caps at 30 points if your project is demonstrated on Week 15 labs, and caps at 20 points if it is during Week 16 labs.